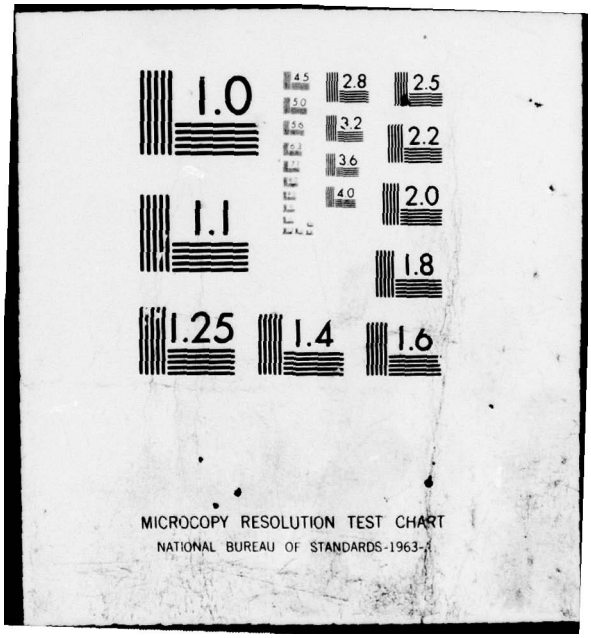
AD-A073 424 UTAH UNIV SALT LAKE CITY DEPT OF COMPUTER SCIENCE F/G 9/2
SURFED - AN INTERACTIVE EDITOR FOR FREE-FORM SURFACES,(U)
MAR 78 R P DUBE, G J HERRON, F F LITTLE N00014-77-C-0157
NL

UNCLASSIFIED

1 OF 1
AD
A073424

END
DATE FILMED
9-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL II

MIL-STD-847A
31 January 1973

(12) 8. S.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
2. GOVT ACCESSION NO
3. RECIPIENT'S CATALOG NUMBER

(6) 4. TITLE (and Subtitle)
SURFED - An Interactive Editor for Free-Form Surfaces

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

(10) 7. AUTHOR(s)
R. Peter Dube, Gary J. Herron, Frank F. Little, Richard F. Riesenfeld

8. CONTRACT OR GRANT NUMBER(s)
N00014 77 C 0159

9. PERFORMING ORGANIZATION NAME AND ADDRESS
University of Utah
Computer Science Department
Salt Lake City, Utah 84112

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research
Dept. of the Navy
Arlington, VA 22217 (Denicoff)

(11) 12. REPORT DATE
MAR. 78

13. NUMBER OF PAGES
(12) 21 p.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Office of Naval Research Resident Repr.
University of California
553 Evans Hall, Berkeley, CA 94720

15. SECURITY CLASS. (of this report)

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
(15) N00014-77-C-0159

18. SUPPLEMENTARY NOTES
NSF-MCS77-13917

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD A073424

DDC FILE COPY

DDC RECEIVED AUG 30 1979 C

79 08 29 102

701949

SURFED - An Interactive Editor for Free-Form Surfaces*

by

R. Peter Dube †

Gary J. Herron ‡

Frank F. Little ‡

Richard F. Riesenfeld ‡

*This work has been supported in part by the National Science Foundation (MCS 74-13017-A01) and the Office of Naval Research (N00014-77-C-0157)

† University of Maine at Orono, Orono, Maine 04473

‡ University of Utah, Salt Lake City, Utah 84112

Abstract

An editor is described for creating and modifying free-form surfaces. A modular system was developed in order to provide the researchers with a facility for communicating design ideas and new mathematical forms through an interactive graphical interface to a computer-based model. To achieve this it was necessary to invent a new graphical construct called a "spider" for inputting 3-dimensional parameters. This experimental system has the essential features of a large-scale implementation, with the capability of utilizing many new surface forms that have not yet been tried in actual applications.

1. Introduction

Interactive surface design is concerned with the problem of specifying, manipulating, viewing, and modifying a computer based mathematical model of a free-form sculptured surface. Since 1964 when Coons introduced the now classical "Coons Patch", many researchers have formulated sophisticated and useful mathematical models for describing curved surfaces. In order to employ these new models for computer aided geometric design, it is essential to provide the designer with a graphical interface that is intuitive, comfortable, and sufficiently powerful, so that a designer can satisfactorily produce a computer model that embodies his ideas. This paper describes the philosophy and structure of a system that has been developed as a research aid for studying the utility and performance of new mathematical surface forms.

II. Requirements of the System

Work on this system was begun because there was a need for members of the Computer Aided Design Group to have a research facility to study and analyze visually the new mathematical surface forms which were being developed within the group, and to compare these with other existing forms. Many of the forms that were being considered were the parametric "patch" type defined locally, i.e. piecewise, over either a square or a triangular domain, $(x_i(u,v), y_i(u,v), z_i(u,v))$. These models all fall into a class called finite dimensional models because they can be fully specified by a matrix of numerical values. A surface is then comprised of a compatible set of patches whose parameters are constrained to insure a "smooth" joint between any two adjoining patches. In the context of this work a "smooth" surface is one that has at least a continuous unit normal.

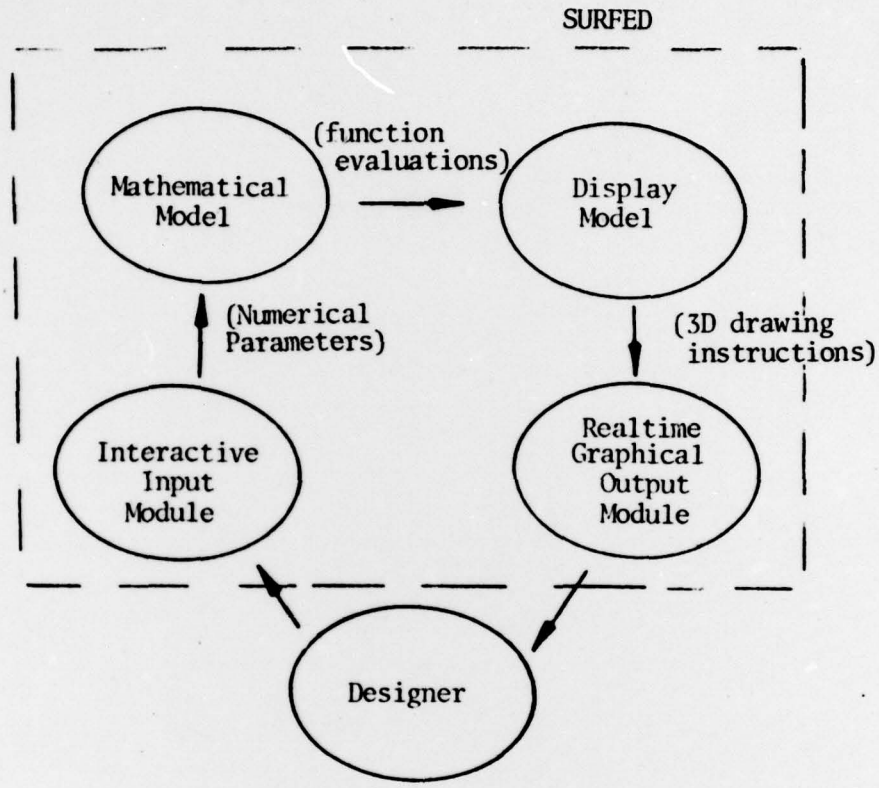
In addition to the requirement that it should be straight-forward to introduce a new surface form into the system, it is a further requirement of the system that the user be able to inspect and modify the model in a convenient and effective manner. Dynamic viewing of the model including simple controls for the viewing motion is a helpful method for achieving the important sense of 3-dimensional space. The input interface has to provide the user with access to parameters of the model so they are natural and understandable. This is especially difficult because the interface must offer the user the facility for communicating 3-dimensional information by means of 1-dimensional and 2-dimensional devices, like knobs and the data tablet. Considerable research has been devoted to the input interface aspect of the computer aided geometric design problem. Most solutions have been highly idiosyncratic and not completely satisfactory, but nonetheless, acceptable to some users.

In the course of discussing, specifying, and implementing the graphical routines that form the user interface of this system, the authors began to view this system as a graphical means of creating and editing a free-form surface, a kind of "surface editor" through which the user can make a well defined geometric statement. The various light menus, in conjunction with the relative settings of the appropriately activated input devices, give the designer access to all of the parameters of the mathematical model so that he can interactively translate a preliminary geometric notion into an acceptable and precise computer model.

III. An Interactive Surface Editor

When the user begins to use the SURFED surface editor he must first select one of the many kinds of surface patches that are available in the system. These basic design elements include the following schemes: Bicubic and 12-parameter Gregory Rectangular Patches, 9-parameter Cubic and various Triangular Coons Patches, B-spline Patches, and Shepard's Scheme for arbitrarily spaced data. The next step is to specify interactively at the vertices of each patch those of the following common parameters that are necessary to define a surface: position, u-tangent, v-tangent, and cross-partials or "twist". The notorious twist, the u partial of the v-tangent and the v partial of the u-tangent, is only necessary for the classical bicubic patch. These options cover the input requirements for the finite dimensional models. In order to define the more general "transfinite" surface models, it is necessary to provide function (vis-a-vis numerical data) input for the boundary curves and possibly for their associated normal derivatives, as for the general Coons patch case.

SURFED is divided into four separate functional modules which are depicted in Figure 1. The first module is the embodiment of the mathematical models available with this system. It performs the necessary function evaluations. The second module requests these evaluations along grid lines and generates a display model. Its function is independent of the particular patch type that has been chosen. Thus the addition of new surface types is a local, restricted, and independent change that does not lead to side effects outside the first module. The third module transforms the 3-dimensional line drawing instructions of the display model into the lines and menus which appear on the screen. This component consists of the realtime graphical routines required to clip, view



Principal Information Flow

Figure 1

and otherwise transfer the 3D display model into a 2D screen image simulating motion in space.

The fourth module, which is devoted to managing the interactive graphical input of surface parameters, is the module whose design is most critical, for through it the designer should feel a comfortable illusion of 3D space by the use of 1-dimensional and 2-dimensional devices. The approach taken in SURFED to solving this input interface problem is to offer to the user graphical "handles", simplified realtime graphical constructs that are intuitive to understand yet complete in the sense that they make available the full parametric freedom of the model.

The computational tasks of this system are divided into two distinct classes: realtime and nonrealtime. In order to attain realtime response to operations that are not directly facilitated by special purpose hardware, we defined considerably simplified models that could be calculated in realtime. This allowed for realtime tablet interaction with the approximation to the model, an approach that gave rise to new graphical handles like the spider discussed in IV. The output viewing calculations are also done in realtime. This is possible in this implementation because there is hardware support for viewing transformations like perspective, rotation, translation, zoom, clipping, and depth cueing. Therefore the designer can modify parameters and receive visualization cues in realtime. The evaluation of the mathematical model is not a realtime operation, although the modification of the parameters is done in realtime. Obviously one cannot expect a complicated model to be calculated in realtime. Often modifications result in only local changes to the model, so that a significant increase in the speed of updating the model can be effected by restricting the recomputation to the region of change.

This system is implemented on an Evans and Sutherland Picture System graphics terminal. The host computer is a PDP 11/45 which includes a floating point processor.

IV. Graphical Handles

Requiring the designer to supply numerical input parameters directly does not, in general, lead to an intuitive scheme for computer aided design. A new graphical handle called a "spider" was developed to aid the user in the process of designing a sculptured surface, (see Figure 2).

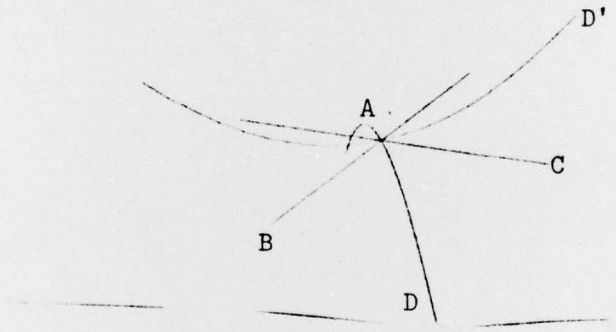


Figure 2. The spider - A: positional information, B: u-partial, C: v-partial, D and D' cross-partials.

The spider is the evaluation of the bilinear Taylor expansion,

$B(u,v) = F_i + u \frac{\partial F_i}{\partial u} + v \frac{\partial F_i}{\partial v} + u\,v \frac{\partial^2 F_i}{\partial u \partial v}$ along the lines in the compass directions, Figure 3.

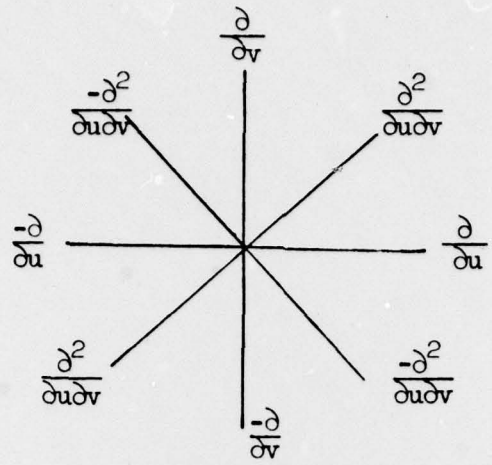


Figure 3. The compass directions $u \in [-1/4, 1/4]$, $v = 0$;
$v \in [-1/4, 1/4]$, $u = 0$;
$u = v \in [-1/4, 1/4]$;
$u = -v \in [-1/4, 1/4]$.

When a modifying mode is selected, spider handles appear on the screen at each patch vertex, Figure 5a. The parameters that define the spider, $\{F_i, \frac{\partial F_i}{\partial u}, \frac{\partial F_i}{\partial v}, \frac{\partial^2 F_i}{\partial u \partial v}\}$, are initially supplied by the parameters of the patch at the associated vertex. The bilinear Taylor expansion is therefore a local approximation to the patch in the neighborhood of the vertex.

As soon as a particular vertex of interest is identified, the other spiders vanish leaving only one sensitized vertex, Figure 5b. The position of the center of the spider determines the intended location of the vertex as the spider is translated in realtime. The 8 legs of the spider control the derivatives $\frac{\partial F}{\partial u}$, $\frac{\partial F}{\partial v}$, and $\frac{\partial F}{\partial u \partial v}$ of the model. Interaction with the spider is accomplished by computing new coefficients for the bilinear polynomial, so that a selected

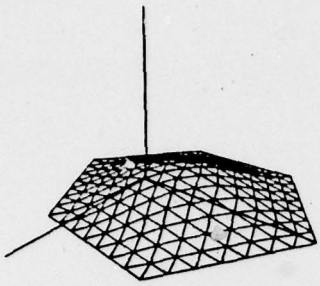
foot is dragged by the pen. When the leg is released, the model is updated by the position, tangents, and twist taken from the spider, Figure 5b and 5c.

The spider, as we have implemented it, serves a dual role. It is not only an attractive design handle, but the changes in the spider's posture forshadows the changes that will actually take place in the surface. Since dynamic update of the surface as the pen moves is not possible, a compromise is reached. The simplicity of the mathematical form of the spider makes it possible to dynamically update the spider as the pen moves. This update is an approximation of what the surface will look like when the complete display model is updated.

V. Implementation

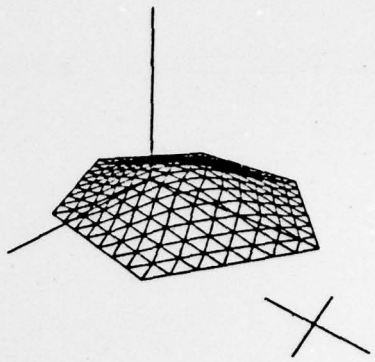
This section will consider the aspects of implementation relative to the following major components: the main routine and overall programming structure, the data structure for modelling and displaying objects, and the graphical editing module.

The general structure of this program uses a main routine that examines a task list and calls in appropriate modules. Control is always returned to the main routine. The task list in this system was chosen to be a stack, and each task of the system has the capability of pushing tasks onto the stack. The highest level menu program, Manual Task Scheduler (MTS) has the characteristic that it always pushes its own name on the stack before returning to Main. Thus MTS is always present on the task stack. When a menu pick is made by the user, MTS places the name of the corresponding subroutine on the stack above its own name. The main menu program, MTS, offers the following functions: Edit,

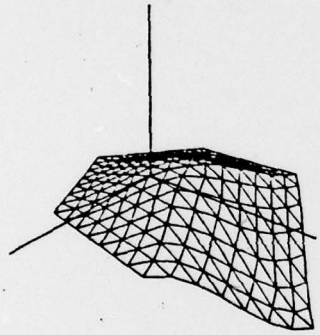


EDIT	IN-OUT	ROTATE X	SCALE
AXIS	JOYSTICK	ROTATE Y	TRANSL TE
RETURN	MORE	ROTATE Z	RESET

a) Original Surface

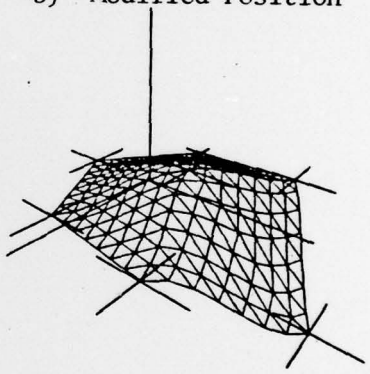


b) Modified Position

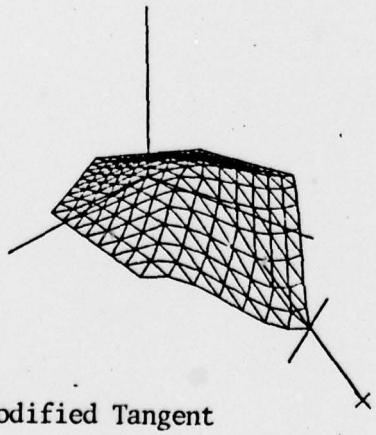


GRAPH	RECOMPUT

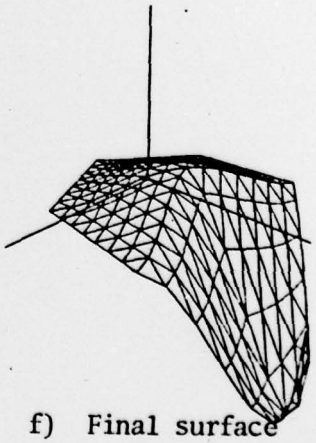
c) Modified Surface



d) Parameter Handles Displayed



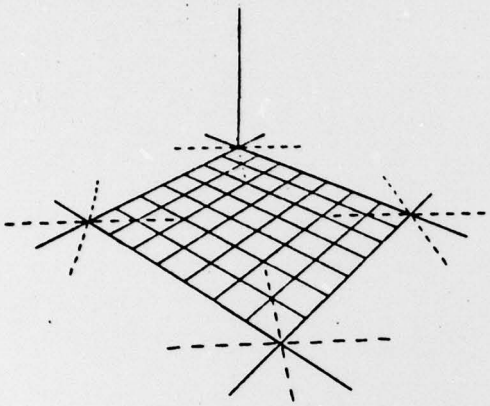
e) Modified Tangent



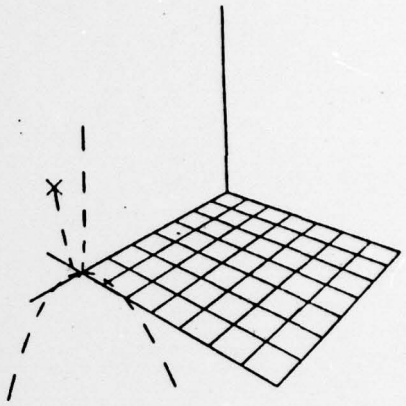
f) Final surface

Editing Session with Triangular Patches

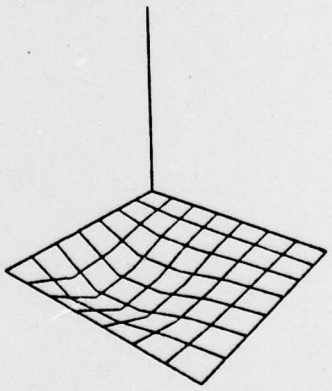
Figure 4



a) Parameter Handles for the Bicubic Patch



b) A New Twist



c) The Resulting Surface

Editing Session with One Bicubic Patch

Figure 5

Update model, Store model, Recall model, Alter view, Change rendering mesh, and Quit. Main pops the stack and invokes the named program. This protocol has the effect of CALL "POP(STACK)". The design of this control structure was clearly influenced by the restriction that Fortran does not allow the name of a subroutine to be passed as a runtime parameter. The stack consists of a common array and a common pointer in the array. This scheme effectively implements a priority queue scheduler, where the most recent request has highest priority.

The data structure we use in this implementation encompasses all the information necessary to generate the graphical (discrete) model. This structure is composed of three arrays: the parameter data array, a vertex array, and a third array which defines the patch list, patch type and meshing instructions, Figure 6.

The data structure used to represent the discrete model has been picked to reflect the Evans and Sutherland Picture System. Adopting this data structure provides a convenient segmentation of the display file into three categories: static, occasional update, and realtime. This segmentation corresponds to the menu, the discritized surface, and the spiders.

The purpose of the Edit module is to alter parameter data. A point is required to appear under a moving cursor location on the screen. The cursor is controlled by a pen and data tablet. A process of hit-testing described below is used to select a point from the control points. The then sensitized point follows the cursor until this mode is terminated by the user.

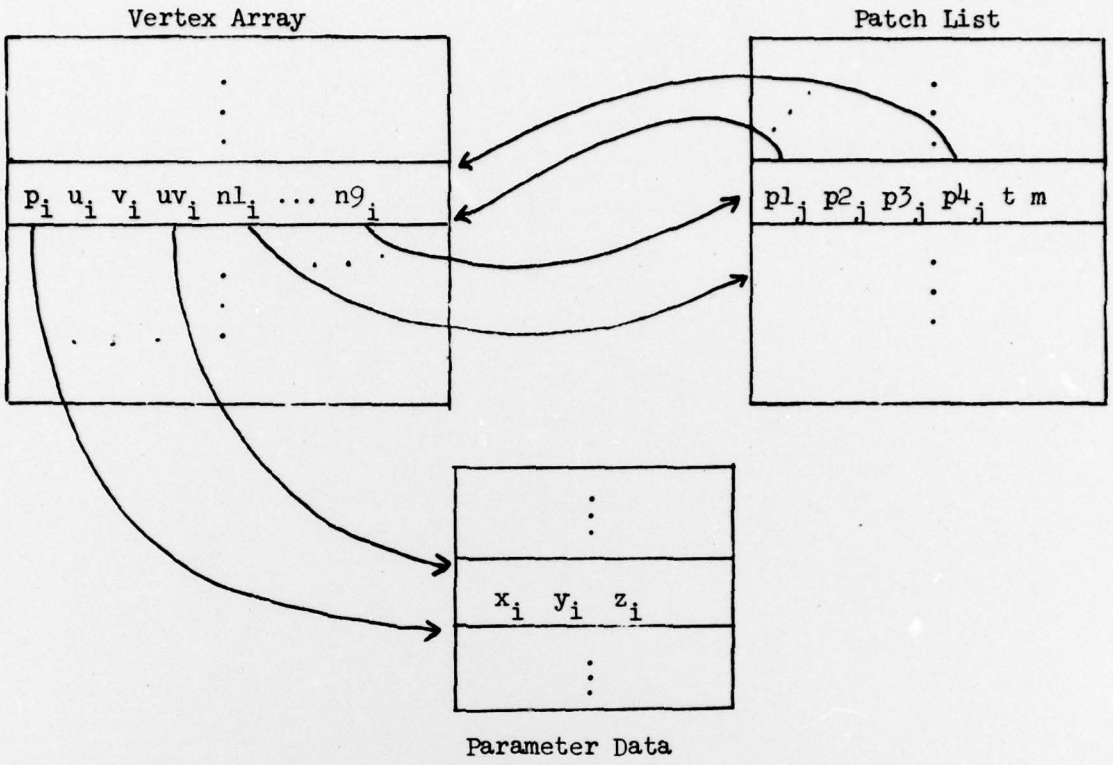


Figure 6. p_i, u_i, v_i, and uv_i point to the position, u-tangent, v-tangent and cross partial associated with the ith vertex. nl_i, ... $n9_i$ are pointers to the patches containing this vertex. pl_j, $p2_j$, $p3_j$ and $p4_j$ are pointers to the vertices which make up the jth patch. The patch type represented by t and m indicates the meshing instruction.

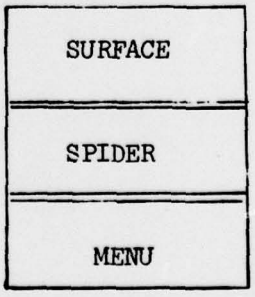


Figure 7. Display file segmentation.

The hit-testing procedure we use is supported in part by the hardware of the Picture System. This support can be described as a logical function Hit. Hit is called with arguments consisting of a parameter data point and a half-width. It will return true if the point is within the specified half-width of the cursor location. This routine is utilized in a loop which decreases the half-width until a unique point is chosen from the control points.

After a parameter controlling point (x, y, z) has been identified its position is maintained under the center of the moving cursor. It is known that the cursor position and the point (x, y, z) are identified under the view transformation. This transformation is accomplished in hardware by the following equation:

$$(hr, hs, ht, h)^T = M\,(x, y, z, 1)^T.$$

M is a four by four matrix which transforms the points according to the user specified simple transformations. After recovering M from the hardware registers the quantity t_{tab} is computed from the transformation equation. The value of t_{tab} remains fixed as (r_{tab}, s_{tab}) is dynamically acquired from the tablet. A linear system is solved to maintain the current (x, y, z) upon each acquisition. The proximity switch of the stylus is used to terminate this dynamic update. The effected patches are then recomputed to update the display file.

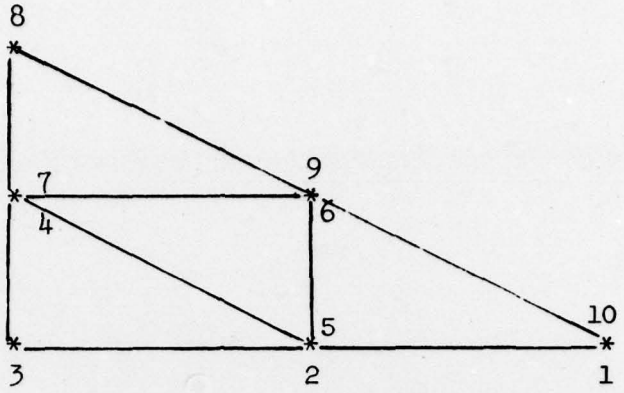


Figure 8. Rendering path for a triangle which avoids redundant evaluations and pen movements. Function evaluations are represented by '*'.

VI. Conclusion

SURFED has the most of essential features of a large scale system. The modular structure provides definition and independence of the logical sub-problems. The use of a spider as a realtime input device has been a successful design handle that offers good graphical cues for what the effect of a modified parameter will be. The spider suggests in realtime what the shape of the modified surface will be. The spider has also provided a solution to the traditionally difficult problem of making the twist parameter understandable to the user.

The division of the computing tasks into those that are realtime and those that are not has been a natural one for the computer configuration that was used. The realtime motion is vital for the understanding of a complicated curvilinear surface. The main computational bottleneck of the system is clearly in the evaluation of the mathematical model. This bottleneck precludes the use of the actual model to communicate parameter modification in an interactive loop. A simplified surface construct is used in a high speed subloop to overcome this problem.

In the long term it is felt that this kind of research will bring the computer into more widespread use as a powerful tool for attacking problems in geometric-design through computer graphics interfaces.

Acknowledgment: The authors wish to thank members of the Computer Aided Geometric Design Group at the University of Utah for their support and cooperation during the development of SURFED.

References

1. Barnhill, R. E., Representation and approximation of surfaces, Mathematical Software III (Edited by J. R. Rice), Academic Press (to appear 1977, pp. 68-119).

2. Coons, S.A., Surfaces for computer-aided design of space forms. MIT Project MAC TR-41 (June 1964).

3. Evans and Sutherland Computer Corporation, The picture system user's manual. Salt Lake City (1976).

4. Gregory, J. A., Piecewise interpolation theory for functions of two variables. Doctoral dissertation, Brunel University (1975).

5. Newell, M. E., The utilization of procedure models in digital image synthesis. Doctoral dissertation, University of Utah (1975).

6. Peters, G. J., Interactive computer graphics application of the parametric bi-cubic surface to engineering design problems. Computer Aided Geometric Design. (Barnhill, R. E. and Riesenfeld, R. F. Editors), Academic Press, London (1974) pp. 259-302.

7. Poeppelmeier, C. C., A Boolean sum interpolant scheme to random data for computer aided geometric design. Master's thesis, University of Utah (1975).

8. Riesenfeld, R. F., Applications of B-splines approximation to geometric problems of computer aided design. Doctoral dissertation, Syracuse University (1973).

9. Riesenfeld, R. F., Aspects of modelling in computer aided geometric design. Proceedings of NCC, Vol. 44, AFIDS Press (1975) pp. 597-602.

10. Shepard, D., A two-dimensional interpolation function for irregularly spaced data. Proceedings of 23rd Association for Computing Machinery National Conference (1968) pp. 517-524.